

CLAIMS

1. A command line utility embodied in one or more computer-readable media, the command line utility comprising:

5 an object model command schema to define a mapping between one or more commands and an object model target schema, the one or more commands generated by the command schema and configured to operate against the target schema through the command line utility.

10 2. A command line utility as recited in claim 1, the command schema comprising:

 an alias class which defines a command template, each alias of the alias class representing a single command.

15 3. A command line utility as recited in claim 2, the command schema further comprising:

 a verb class as a subclass to the alias class, each instance of the verb class representing a behavior available through an alias;

20 a format class as a subclass to the alias class, each instance of the format class representing a list of properties that will be returned through processing an alias; and

 a connection class as a subclass to the alias class, each instance of the connection class providing parameters used through an alias when establishing a connection to a target namespace within the target schema.

4. A command line utility as recited in claim 3, the command schema further comprising:

a parameter class as a subclass to the verb class, each instance of the parameter class representing parameters accepted by a verb from the verb class.

5

5. A command line utility as recited in claim 3, the command schema further comprising:

a property class as a subclass to the format class, each instance of the property class representing property values for properties within a list of
10 properties.

CONTINUATION SHEET

6. A command line utility as recited in claim 2, the command schema further comprising:

a localized string class, each instance of the localized string class
15 providing language-specific text relevant to an alias.

7. A command line utility as recited in claim 2, the command schema further comprising:

a qualifiers class, each instance of the qualifiers class representing
20 qualifiers to be applied to elements of an alias.

8. A command line utility as recited in claim 2, the command schema further comprising:

a see-also association, each instance of the see-also association
25 providing associations to aliases which are related to an alias.

9. A command line utility as recited in claim 1, wherein the one or more commands are organizable into role-oriented namespace, such that commands related to particular administrative tasks are found together.

5

10 10. A command line utility as recited in claim 1, wherein the command schema is extensible to permit the generation of additional commands to be added to the one or more commands.

11. A command line utility as recited in claim 1, wherein the command schema is configurable to permit reconfiguration of the one or more commands.

12. A command line utility as recited in claim 1, the command schema comprising:

15 an alias class to define alias instances, each alias instance representing a single command;

a verb class to define verb instances, each verb instance representing behavior available through an alias instance;

a parameter class to define parameters accepted by a verb instance;

20 a format class to define format instances, each format instance having a list of properties to be displayed through an alias instance;

a property class to define property instances, each property instance representing a property value from a property list;

DRAFT 2018-02-06

a connection class to define connection instances, each connection instance representing connection parameters used by an alias instance to establish a connection to the target schema;

5 a qualifier class to define qualifier instances, each qualifier instance representing constraints on elements of an alias instance;

 a localized string class to define localized string instances, each localized string instance representing a text localization for translating text into a localized language; and

10 a see-also association to associate an alias instance with other related alias instances.

13. A command line utility as recited in claim 1, wherein the target schema comprises a WMI object model.

15 14. A command line utility as recited in claim 1, wherein the command schema is a plurality of command schemas, each command schema representing an instant set of one or more commands.

20 15. A command line utility as recited in claim 1, wherein the command schema resides on a local machine and the target schema resides on one or more remote machines.

16. An object model schema embodied in one or more computer-readable media, the object model schema comprising:

an alias class to define alias instances, each alias instance representing a command;

a verb class to define verb instances, each verb instance representing behavior available through an alias instance;

5 a parameter class to define parameters accepted by a verb instance;

a format class to define format instances, each format instance having a list of properties to be displayed through an alias instance;

a property class to define property instances, each property instance representing a property value from a property list;

10 a connection class to define connection instances, each connection instance representing connection parameters used by an alias instance to establish a connection to the target schema;

a qualifier class to define qualifier instances, each qualifier instance representing constraints on elements of an alias instance;

15 a localized string class to define localized string instances, each localized string instance representing a text localization for translating text into a localized language; and

a see-also association to associate an alias instance with other related alias instances.

20

17. A management application embodied in one or more computer-readable media, the management application comprising:

a set of commands generated by an object model command schema to operate against an object model target schema, the command schema defining a mapping between the set of commands and the target schema; and

an interface utility to facilitate implementation of individual commands within the set of commands.

18. A management application as recited in claim 17, wherein the
5 interface utility is a command line interface utility.

19. A management application as recited in claim 17, wherein the interface utility is a graphical user interface utility.

10 20. A management application as recited in claim 17, wherein the set of commands is organizable into role-oriented namespace, such that individual commands related to particular administrative tasks are found together.

15 21. A management application as recited in claim 17, wherein the command schema is extensible to permit the generation of additional commands to be added to the set of commands.

20 22. A management application as recited in claim 17, wherein the command schema is configurable to permit reconfiguration of individual commands in the set of commands.

23. A management application as recited in claim 17, wherein the command schema comprises:

25 an alias class to define alias instances, each alias instance representing a command within the set of commands;

□ ◻ ◉ ◇ ◆ ◇ ◆ ◉ ◻ □

- a verb class to define verb instances, each verb instance representing behavior available through an alias instance;
- a parameter class to define parameters accepted by a verb instance;
- a format class to define format instances, each format instance
- 5 having a list of properties to be displayed through an alias instance;
- a property class to define property instances, each property instance representing a property from a property list;
- a connection class to define connection instances, each connection instance representing connection parameters used by an alias instance to establish
- 10 a connection to the target schema object model;
- a qualifier class to define qualifier instances, each qualifier instance representing constraints on elements of an alias instance;
- a localized string class to define localized string instances, each localized string instance representing a text localization for translating text into a
- 15 localized language; and
- a see-also association to associate an alias instance with other related alias instances.

24. A management application embodied in one or more computer-readable media, the management application comprising:

- a first object model to control the configuration and behavior of the management application in operating against and managing a second object model.

25. A management application as recited in claim 24, wherein the first object model comprises:

an alias class to define alias instances, each alias instance representing a command used by the management application;

5 a verb class to define verb instances, each verb instance representing behavior available through an alias instance;

a parameter class to define parameters accepted by a verb instance;

a format class to define format instances, each format instance having a list of properties to be displayed through an alias instance;

10 a property class to define property instances, each property instance representing a property from a property list;

a connection class to define connection instances, each connection instance representing connection parameters used by an alias instance to establish a connection to the second object model;

15 a qualifier class to define qualifier instances, each qualifier instance representing constraints on elements of an alias instance;

a localized string class to define localized string instances, each localized string instance representing a text localization for translating text into a localized language; and

20 a see-also association to associate an alias instance with other related alias instances.

26. A management application as recited in claim 24, wherein the second object model comprises a WMI object model.

27. A method comprising:
receiving a command through a command line interface;
fetching an alias for the command;
interpreting the command based on the alias and the current
5 operating environment of the command line interface;
executing the command as one or more WMI API calls against a
target namespace;
receiving WMI data in XML form;
applying an XSL style sheet format the WMI data; and
10 presenting the WMI data through the command line interface.

28. A method as recited in claim 27, wherein the alias is an instance of
an alias class.

15 29. A method as recited in claim 28, wherein the alias and the current
operating environment are determined by command entries to the command line
interface.

20 30. A method as recited in claim 28, wherein the alias class is the
primary class in a command schema.

31. A method as recited in claim 30, wherein the command schema
additionally includes:

25 a verb class to define verb instances, each verb instance representing
behavior available through an alias;

P
A
T
E
N
T
U
R
E

15

20

30

35

40

45

50

55

60

65

70

75

80

85

90

95

a parameter class to define parameters accepted by a verb instance;

a format class to define format instances, each format instance having a list of properties to be displayed through an alias;

a property class to define property instances, each property instance representing a property from a property list;

a connection class to define connection instances, each connection instance representing connection parameters used by an alias to establish a connection to the target namespace;

a qualifier class to define qualifier instances, each qualifier instance representing constraints on elements of an alias;

a localized string class to define localized string instances, each localized string instance representing a text localization for translating text into a localized language; and

a see-also association to associate an alias with other related aliases.

32. One or more computer-readable media storing computer-executable instructions that, when executed on one or more processors, perform the method of claim 27.

33. A method of managing objects in a target schema comprising:

providing a user interface;

defining a command structure through an object-oriented command schema, the command schema including an alias class;

instantiating an object of the alias class as an alias by receiving parameters of the alias class through the user interface, the alias representing a command which maps to an object in the target schema; and

executing the command against the object in the target schema.

5

34. A method as recited in claim 33, further comprising:
receiving data related to the object in the target schema in XML

form;

retrieving an XSL file from a location defined through the alias;

10 formatting the data for display by applying an XSL style sheet from
the XSL file; and

displaying the data through the user interface.

35. A method as recited in claim 33, wherein the user interface is a
15 command line user interface.

36. A method as recited in claim 33, wherein the user interface is a
graphical user interface.

20 37. One or more computer-readable media storing computer-executable
instructions that, when executed on one or more processors, perform the method of
claim 33.